

### **REMARKS/ARGUMENTS**

The Office Action dated January 25, 2005 has been received and carefully considered. Claims 1-3, 5-12, and 14-25 are pending in the application. Claims 1, 10, and 21 have been amended. Claims 22-25 are cancelled. New claims 26-29 are added. No new matter is added by this Amendment. Applicants believe that the application is now in condition for allowance and notice thereof is respectfully requested.

#### **Pending Rejections**

Claim 1 is objected to due to an informality noted by the Examiner.

Claims 1, 10, 19 and 21 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Claims 1, 3, 6-10, 12 and 15-21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,918,232 to Pouschine *et al.* ("Pouschine") in view of U.S. Patent No. 5,584,024 to Schwartz ("Schwartz").

Claims 2, 5, 11 and 14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pouschine in view of U.S. Patent No. 6,574,623 to Leung ("Leung").

Claims 22-25 are objected to as being dependent upon a rejected base claim but otherwise allowable if presented in independent form including all limitations of the base claims and any intervening claims.

#### **INFORMALITY OF CLAIM LANGUAGE**

Applicants have amended Claim 1 to correct the informality noted by the Examiner.

REJECTIONS UNDER 35 U.S.C. § 101

Applicants believe the previously presented Claims 1, 10, 19 and 21, either by themselves or read in light of the Specification, did fall within the technological arts of computer databases. Nevertheless, Applicants have amended Claims 1, 10, and 21 to further clarify their computer-implemented nature. In addition, Claim 19 is directed to a processor-readable medium which is clearly within the technological arts. Accordingly, Applicants submit this rejection has been overcome and respectfully request that the rejection be withdrawn.

ALLOWABLE SUBJECT MATTER

Applicants are grateful that the Examiner considers the previously presented Claims 22-25 allowable. Per the Examiner's suggestion, Claims 22-25 have been rewritten and presented in independent form as New Claims 26-29. Accordingly, Claims 26-29 are now in condition for allowance.

REJECTIONS UNDER 35 U.S.C. §103(a)

The Office Action rejects all the pending claims for obviousness based on Pouschine in view of either Schwartz or Leung. Applicants respectfully submit that the cited references, alone or in combination, do not teach or suggest all the elements in the claimed invention.

The Examiner quotes the following passages from Pouschine as allegedly disclosing "*a query structure assembly module for defining a query structure based upon a plurality of query assembly rules and a desired data set*":

The method includes the steps of inputting a query, parsing the query; creating a query component tree, inputting model metadata into the Domain Modeling Rule Set Preparation Module, generating a Domain

Modeling Rule Set from the Domain Modeling Rule Set Preparation Module, inputting the Domain Modeling Rule Set into the calculation engine, and generating an execution tree with rules from the calculation engine. (Col. 4, lines 61-67)

The method 200 starts as a Hyperstructure Query Language (HQL) query 202 which is passed to a parser 122 which converts the HQL text to a query component tree 204 which represents the component parts of the query 202. (Col. 16, lines 23-26)

Applicants respectfully submit that the quoted passages (or the rest of the Pouschine reference for that matter) fail to show a query structure assemble module based on “query assembly rules.” It should be noted that Pouschine does not use the term “rule” in the same sense as in the present application. In the present application, “query assembly rules” are typically rules for evaluating the desired data set and for generating an optimized query execution plan. Page 13, lines 14 - 17. The query assembly rules may embody functions of the base selection module 318 (page 15, lines 11-13), the intermediate table selection module 320 (page 16, lines 3-5), the intermediate table method module 322 (page 16, lines 22-23), and/or the join path selection module 324 (page 17, lines 22-23), for example. That is, the query assembly rules may comprise a number of logical rules, dependencies and conditions which are utilized by the query structure assembly module 312 during initial parsing of the desired data set parameters and assembly of the query structure. Page 15, lines 9-15. In other words, the query assemble rules are rules that govern the query structure assembly process.

In Pouschine, the term “rule” is explained in the following passages:

Model--A named, file-like package consisting of:

- \* Dimensions, elements, and rules
- \* Blobs to store third party data and worksheets
- \* A set of user entered values
- \* Information spaces (Col. 8, lines 1-5)

To obtain the values for cells in the model, DOLAP uses two types of rules: 1) formula rules which tell DOLAP to compute the cell value mathematically from other cells in the model, for example, Revenue is calculated by multiplying Price times Units Sold; and 2) data map rules which tell DOLAP to retrieve the cell value from the database. (Col. 12, lines 1-6)

Domain Modeling Rule Set Preparation is the process of gathering all of the rules from the model, turning them into valid actions that can be used by the calculation engine, and prioritizing them so that the proper rules take precedence. (Col. 17, lines 11-14)

See also, col. 12, lines 9-50, for a more detailed description of “formula rules” and “data map rules.” At least two conclusions can be readily drawn from the above-quoted passages: (1) the rules in Pouschine come from user-defined models; and (2) these rules are either formulae for computing the cell value or data maps for retrieving the cell value. In other words, Pouschine’s “rules” are essentially the query structure itself, rather than rules that govern the assembly of the query structure. Therefore, despite the confusing use of a similar word, Pouschine does not disclose “query assembly rules” as recited in claims 1, 10 and 21.

The Office Action acknowledges that Pouschine “does not explicitly indicate in the first limitation of the current claim [Claim 1] the step of basing the defining of a query structure on a plurality of query assembly rules.” Office Action, page 7. However, the Examiner points to the following passage as allegedly disclosing this limitation:

These are provided to Domain Modeling Rule Set Preparation 208 which generates the domain modeling rule set 126, which are then supplied to the calculation engine 18, which takes the applicable rules and adds them to the query tree 204 to produce an execution tree with rules 212 for the query engine 132. (Col. 16, lines 30-35, emphasis the Examiner’s)

As discussed above, the “Domain Modeling Rule Set” in Pouschine is essentially user-defined formulae and/or actions for computing or retrieving the cell value. Since the “[calculation

engine 18] takes the applicable rules and adds them to the query tree 204,” the Domain Modeling Rule Set is part of the query structure itself. Therefore, the Domain Modeling Rule Set cannot be the functional equivalent of the “query assembly rules” which govern the assembly of query structure.

The Examiner also quotes the following passages from Pouschine as allegedly disclosing the limitation of “*the query assembly rules being used by the query structure assembly module to evaluate the desired data set*”:

a Domain Modeling Rule Set Preparation Module, a query engine, and an evaluator which communicates with an SQL generator ... (Col. 4, lines 57-58, emphasis the Examiner’s)

The underlining of “Rule Set” confirms that there may be some confusion about the meaning of “Rule Set” in the Pouschine reference and the “query assembly rules” in the present application.

Further, the “evaluator” in Pouschine does not “evaluate the desired data set” as claimed in the present application. Rather, the evaluator 128 “decides whether further data is required from a relational database. If further data is required, evaluator 128 communicates with a Relational DataBase Management System (RDBMS) 216 through an SQL generator 218. Evaluator 128 can also communicate with a math library 220, if a calculation is required, or a sorting and processing system 222, if the process requires ordering of results or sorting in some manner.” Col. 16, lines 38-46. As such, Pouschine’s evaluator 128 is more like an intermediary between the query engine and the relational database. See Figure 8.

The Examiner further quotes the following passages from Pouschine as allegedly disclosing *"a syntax assembly module for defining at least one query language statement based upon the defined query structure"*:

The method 200 starts as a Hyperstructure Query Language (HQL) query 202 which is passed to a parser 122 which converts the HQL text to a query component tree 204 which represents the component parts of the query 202. (Col. 16, lines 23-26)

However, the quoted passage only pertains to the *parsing* of a HQL text but does not teach the *assembling* of a query structure.

The Examiner quotes the following passages from Pouschine as allegedly disclosing *"a process optimization module for evaluating processing options based upon a database schema associated with the data source"*:

An incoming client request is routed to an HQL parser 122 which transforms the HQL query from its textual form into an internal representation. After the internal representation of the query is made, the calculation engine 18 looks in the query results cache 124 to see if the query (or any portion of it) can be retrieved from the cache in order to save processing time.

A Domain Modeling Rule Set 126 receives input from both the model 50 and the schema 104, and provides input to the rule evaluator 128, which then communicates with the execution plan 130 and the query engine 132. The query engine 132 creates SQL queries from HQL queries and optimizes these SQL queries by combining them (whenever possible) and also optimizes the performance of the calculation engine 18 by performing as many calculations as possible with SQL statements executed by the database. (Col. 15, lines 51-66)

This is followed by inputting the execution tree with rules to the query engine, generating an optimized execution tree from the query engine, inputting the optimized execution tree to the evaluator, and communicating between the evaluator and any of a number of data reference devices. An execution tree with results is generated, the query results are packaged and the results displayed in an output device. (Col. 5, lines 1-5)

Although the Pouschine reference briefly alludes to the execution tree being optimized, the disclosure is very limited. Other than combining the SQL queries whenever possible, the Pouschine reference does not teach or suggest any method for process optimization. Nor does Pouschine disclose "evaluating processing options." Further, as conceded in the Office Action, Pouschine does not disclose "*an intermediate data processing method module for evaluating a plurality of methods for generating intermediate data sets within the data source(s).*"

The Examiner cites Schwartz as allegedly disclosing "an intermediate data processing method module for evaluating a plurality of methods for generating intermediate data sets within the data source(s)." However, a close reading of the Schwartz reference reveals otherwise.

First, Schwartz does not deal with process optimization at all. The primary and only concern of Schwartz is to prohibit the selection of semantically incorrect query parameters. See Title and Abstract of Schwartz. Therefore, Schwartz only distinguishes two types of queries, those that are semantically correct versus those that are semantically incorrect. Schwartz does not improve or optimize the efficiency of the query structure by evaluating multiple process options or multiple methods for generating intermediate data sets. Indeed, the text of the Schwartz reference does not even include such terms as "optimize," "evaluate," "efficient," or variations thereof.

Second, the intermediate query language referred to in Schwartz is completely different from the "intermediate data sets" as recited in claims 1, 10, 19 and 21. Schwartz's intermediate query language is a translation of the user's query input from a form of menu selections to a form of easy-to-understand sentences. Col. 9, lines 45-47. In other words, Schwartz's intermediate query language is no more than an alternative presentation of the user input. In the present

application, the “intermediate data sets” refer generally to data calculated or otherwise generated for the purpose of returning the desired data set. See page 15-17 for a description of the intermediate table selection module and the intermediate table method module. Unlike Schwartz’s intermediate query language, the intermediate data sets in the present application are not user inputs, but data generated *in response to* user inputs.

Third, Schwartz does not teach or suggest reusability of intermediate data sets as recited in claims 3, 12 and 19. Applicants agree with the Examiner that Schwartz teaches reusing its English-like queries as shown in FIG. 3B. However, reusing the frequently selected user queries is fundamentally different from reusing intermediate data sets to generate more efficient SQL language. The former is performed by a human user through menu selections while the latter is performed by machine intelligence, specifically the intermediate data processing method module.

In view of the foregoing, the § 103(a) rejections of the independent claims cannot stand. Therefore, the basis is moot for rejecting dependent claims 2, 5, 11 and 14 based on Pouschine further in view of Leung.

Applicants respectfully submits that in view of these amendments and the remarks expressed above regarding the rejections under § 103(a), claims 1, 10, 19, 21, and 26-29 are now allowable over the cited art of record. Each of the remaining claims depends from either claim 1, 10 or 19, and is therefore allowable for at least the reasons expressed above.



**CONCLUSION**

Since the cited references, taken either singly or in combination, fail to teach or suggest the combinations set forth in the pending claims, and further fail to provide any motivation or suggestion of the desirability of modifying the structures or methods to arrive at the claimed combinations, Applicant submits that the pending claims are allowable over the cited references. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections, allow the pending claims and pass the application to issue.

If the Examiner believes that a telephone conference or interview would advance prosecution of this application in any manner, the undersigned stands ready to conduct such a conference at the convenience of the Examiner.


If there are any fees due which are not enclosed herewith, including any fees required for extension of time under 37 C.F.R. §1.136, please charge such fees to our Deposit Account No. 50-0206.

Respectfully submitted,

HUNTON & WILLIAMS LLP

Dated: April 22, 2005  
Hunton & Williams, L.L.P.  
Intellectual Property Department  
1900 K Street, N.W.  
Suite 1200  
Washington, D.C. 20006-1109  
(202) 955-1500 (telephone)  
(202) 778-2201 (facsimile)

By:

  
\_\_\_\_\_  
Brian M. Buroker  
Registration No. 39,125